Model Selection for Neural Network Classification

Herbert K. H. Lee, Duke University
Box 90251, Durham, NC 27708, herbie@stat.duke.edu

June 2000

Abstract

Classification rates on out-of-sample predictions can often be improved through the use of model selection when fitting a model on the training data. Using correlated predictors or fitting a model of too high a dimension can lead to overfitting, which in turn leads to poor out-of-sample performance. I will discuss methodology using the Bayesian Information Criterion (BIC) of Schwarz (1978) that can search over large model spaces and find appropriate models that reduce the danger of overfitting. The methodology can be interpreted as either a frequentist method with a Bayesian inspiration or as a Bayesian method based on noninformative priors.

Key Words: Model Averaging, Bayesian Random Searching

1 Introduction

Neural networks have become a popular tool for classification, as they are very flexible, not assuming any parametric form for distinguishing between categories. Applications can be found in both the frequentist and Bayesian literature. An aspect which has not been thoroughly addressed is model selection. Just as is the case for linear regression, using more explanatory variables may give a better fit for the data, but may lead to overfitting and bad predictive performance. Similarly, increasing the size of a neural neural network may lead to better fits on training data, but may result in overfitting and poor predictions. Thus one needs a method for deciding how to choose a best model, or best set of models. In a larger problem, one also needs a way of searching the model space to find this best model, as it may be impossible to try fitting all possible models. This paper is meant to address these issues.

There are a number of other papers which look at the problem of selecting the optimal size of a neural network. Much of the recent work has been in the Bayesian framework, and includes gaussian approximations for the posterior to approximate posterior probabilities (MacKay, 1992), and reversible jump MCMC methods (Müller and Rios Insua, 1998 (for regression), Andrieu et al., 1999 (for radial basis networks)). More established methods (see Bishop (1995)) include cross-validation (Stone, 1974) and penalized likelihood methods using the Akaike Information Criterion (AIC) (Akaike, 1974), the Bayesian Information Criterion (BIC) (Schwarz, 1978), or the Network Information Criterion (NIC) (Murata et al., 1994). Cross validation has also been applied to variable selection, but not during a selection of model size as well. A Bayesian approach to model selection is Automatic Relevance Detection (ARD) (MacKay, 1994; Neal, 1996), which uses an additional layer of hyperparameters to try to shrink unimportant variables. However, ARD does not allow one to compute posterior probabilities of individual models. The methods of this paper allow one to simultaneously search the model space over both the size of the network and over subsets of explanatory variables. Furthermore, if a Bayesian interpretation is taken, one can estimate posterior probabilities of individual models.

This paper focuses on feedforward neural networks with a single hidden layer of units with logistic activation functions, but the results are clearly generalizable to other flavors of neural networks. The mathematical details of these models are given in Section 2. Section 3 deals with the issues of searching the model space and selecting a best model. Within the Bayesian framework, one may also want to use model averaging for prediction, and this is discussed as well. Bayesian Random Searching (BARS) is introduced as a method that fits within both the frequentist and Bayesian frameworks, and is applicable to many problems including neural networks. Finally, two examples are given.

2 Neural Networks

The particular form of the model used in this paper is derived from the regression context. The regression model can be viewed as using a set of logistic functions as an approximate basis for the space of continuous functions. Denote the multivariate response variable as $\mathbf{y}$, where y_{ig} is the gth component of the ith observation of the response variable, $g \in \{1, \ldots, q\}$, $i \in \{1, \ldots, n\}$. Let x_{ih} be the hth component of the ith observation of the explanatory variable, $h \in \{1, \ldots, r\}$. Then the regression model for y_{ig} is

$$\begin{aligned} y_{ig} &= \beta_{0g} + \sum_{j=1}^{k} \beta_{jg} \Psi_j(\boldsymbol{\gamma}_j^t \mathbf{x}_i) + \varepsilon_{ig} \\ \Psi_j(\boldsymbol{\gamma}_j^t \mathbf{x}_i) &= \frac{1}{1 + \exp\left(-\gamma_{j0} - \sum_{h=1}^{r} \gamma_{jh} x_{ih}\right)} \\ \varepsilon_{ig} &\overset{iid}{\sim} N(0, \sigma^2) \end{aligned}$$

where j is the index on the basis functions, known as *hidden nodes*, the γ's are the coefficients (*weights*) from the explanatory variables to the hidden nodes, and the β's are the coefficients from the hidden nodes to the predicted responses. Ψ_j is the jth basis function, a logistic transformation of a linear combination of the explanatory variables. The fitted values are seen to be a linear combination of the fitted basis functions, and conditional on the basis functions, the fitted value is the regression fit on the basis functions. A common variance is assumed here, but this could easily be generalized to a separate variance for each component of the response vector.

Neural networks of this form have been shown (Cybenko, 1989; Funahashi, 1989; Hornik et al., 1989) to be able to approximate any continuous function arbitrarily well when sufficiently many hidden nodes are used. In the Bayesian context, the posterior is consistent (Lee, 2000). These properties make neural networks a good method for nonparametric regression, in that one does not have to choose a particular parametric form for the model. A practical consequence is the need to choose an optimal number of hidden nodes. This problem is discussed in Section 3.

The key to extending this model to classification is to express the categorical response variable as a vector of indicator variables. The outputs of the neural network are transformed to a probability scale and a multinomial likelihood is used. Let t_i be the categorical response (the *target*) with its value being the category number to which case i belongs, $t_i \in \{1, \ldots, q\}$, where q is the number of categories. Let $\mathbf{y}_i$ be a vector in the alternate representation of the response (a vector of indicator variables), where $y_{ig} = 1$ when $g = y_i$ and zero otherwise. Let p_{ig} be the (true) underlying probability that $y_{ig} = 1$. Denote the fitted estimate for p_{ig} by $\hat{p}_{ig}$. To get this fitted estimate, the continuous valued output of the neural network (denoted $\hat{w}_{ig}$) is transformed to the probability scale. Then the likelihood is

$$f\left(\mathbf{y}|\mathbf{p}\right) = \prod_{i=1}^{n} f\left(t_i | p_{i1}, \ldots, p_{iq}\right) \tag{1}$$

$$f\left(t_i | p_{i1}, \ldots, p_{iq}\right) \propto \left(p_{i1}\right)^{(y_{i1})} \cdots \left(p_{iq}\right)^{(y_{iq})}$$

and the fitted probabilities are found from the neural network by

$$\hat{p}_{ig} = \frac{\exp\left(\hat{w}_{ig}\right)}{\sum_{h=1}^{q} \exp\left(\hat{w}_{ih}\right)}$$

$$\hat{w}_{ig} = \beta_{0g} + \sum_{j=1}^{k} \beta_{jg} \Psi_j(\boldsymbol{\gamma}_j^t \mathbf{x}_i)$$

$$\Psi_j(\boldsymbol{\gamma}_j^t \mathbf{x}_i) = \frac{1}{1 + \exp\left(-\gamma_{j0} - \sum_{h=1}^{r} \gamma_{jh} x_{ih}\right)}$$

In practice, only the first $q - 1$ elements of y are used in fitting the neural network so that the problem is of full rank. $\hat{w}_{iq}$ is set to zero for identifiability of the model.

The parameters of this model are k (the number of hidden nodes), β_{jg} for $j \in \{0, \ldots, k\}$, $g \in \{1, \ldots, q-1\}$, and γ_{jh} for $j \in \{1, \ldots, k\}$, $g \in \{0, \ldots, r\}$. For a fixed network size (fixed k), there are $k*(q+r)+1$ parameters in the model.

This model is frequently referred to as the *softmax* model in the field of computer science (Bridle, 1989). This method of reparameterizing the probabilities from a continuous scale can be found in other areas of statistics as well, such as generalized linear regression (McCullagh and Nelder, 1983) .

In the frequentist setting, this model can be fit using standard numerical optimization routines. In the computer science literature, backpropagation, a gradient descent type algorithm, is popular. In the Bayesian context, one can use either a hierarchical prior, or a noninformative prior, and the model can be fit using Markov chain Monte Carlo.

3 Model Selection

Model selection for a neural network entails both a selection of the structure of the network (how many hidden nodes) and a selection of which explanatory variables to use. Using more hidden units will give a better fit. With sufficiently many units, a network can fit the data perfectly. Thus one needs to choose a proper size for the network, large enough that it will fit well, but small enough to minimize over-fitting and improve predictive performance. At the same time, one must choose the optimal subset of explanatory variables. Just like the case of multiple linear regression over a wide range of candidate explanatory variables, one wants to use only the important variables and to leave the others out of the model. Neural networks present the additional challenge of simultaneously choosing both the variables and the size of the network.

There are many competing ideas on how to choose the best model. Some are based on maximizing the likelihood subject to some penalty, such as the Akaike Information Criterion (Akaike, 1974) and the Bayesian Information Criterion (Schwarz, 1978). Other methods attempt to use part of the data to check the model in some way, such as cross-validation (Stone, 1974). The methodology of this paper originates from the Bayesian approach of choosing the model with highest posterior probability, although as will be explained later, the Bayesian structure is not necessary for implementation or interpretation.

The Bayesian approach also allows model averaging for increased predictive performance and better accounting of uncertainty. In some cases (for example, the heart attack data in Raftery (1996)), more than one model may have high posterior probability, and these models will give different predictions. Using predictions from only a single model will grossly underestimate the variability of the estimate, since it ignores the fact that another model with significant posterior probability made a different prediction. Instead, one should calculate predictions by using a weighted average over all models in the search space, where the weights are the posterior probabilities of the models (Leamer, 1978; Kass and Raftery, 1995). Let y be the response variable, D the data (the observed values of the explanatory and response variables), and M_i the models of interest for $i \in \mathcal{I}$. Then the posterior predictive distribution of y is

$$P(y|D) = \sum_{i \in \mathcal{I}} P(y|D, M_i) P(M_i|D) \,,$$

where $P(y|D, M_i)$ is the marginal posterior predictive density given a particular model (with all other parameters integrated out), and $P(M_i|D)$ is the posterior probability of model i.

The posterior probabilities of the models, $P(M_i|D)$, while conceptually straightforward, turn out to be a computational challenge to estimate. By Bayes' Theorem:

$$P(M_i|D) = \frac{P(D|M_i)P(M_i)}{\sum_j P(D|M_j)P(M_j)} \,.$$

where $P(D|M_i) = \int f(\mathbf{y}|\mathbf{p}) f(\mathbf{p})$ is the marginal probability of the data, which is the likelihood (from Equation 1) times the prior integrated over the parameter space for that model (the β_{jg} and γ_{jh} as described in Section 2). This integral is analytically intractable in the case of a neural network. The (exponentiated) BIC is a useful approximation to the posterior probabilities of the models, as the BIC is an approximation to the log of the Bayes factor for comparing that model to the null model (Schwarz 1978). Recall that the

BIC for model i is defined as

$$BIC_i = L_i - \frac{1}{2} p_i \log n,$$

where L_i is the maximum of the log-likelihood, n is the sample size, and p_i is the number of parameters in model i. It has been shown that $BIC_i = \log P(D|M_i) + O_P(1)$ (see, for example, Kass and Wasserman (1995)). While this may not appear to be a great approximation, the BIC has been shown to be asymptotically consistent for model selection for many classes of models, including mixture models (Keribin, 1997), which share many similarities to neural networks. The BIC approximation has also been found to work well in practice. More details on the difficulties of estimating these normalizing constants for neural networks, as well as on the usefulness of the BIC for this problem can be found in Chapter 4 of Lee (1998).

In the absence of additional information on model size, one can use a noninformative prior that puts equal mass on each model, i.e. $P(M_i) = P(M_j)$ for all i and j. The BIC approximation is then

$$P(M_i|Y) = \frac{P(Y|M_i)}{\sum_j P(Y|M_j)} \approx \frac{e^{BIC_i}}{\sum_j e^{BIC_j}} \,. \tag{2}$$

If one did have information on model size, an informative prior could be used for the model space just as easily. The Bayesian approach automatically finds a balance between improving the fit and not overfitting, because adding additional variables or nodes that do not sufficiently improve the fit will dilute the posterior, causing a lower posterior probability for the model. This approach, in addition to being conceptually straightforward, also has the advantage that it can be used simultaneously for choosing the explanatory variables and the number of hidden nodes.

3.1 Bayesian Random Searching

For a small number of explanatory variables under consideration, one could try fitting models using all possible subsets. However this quickly becomes difficult as the number of variables increases. Furthermore, one would need to select an optimal network size for each subset. Thus an efficient algorithm is necessary for searching over the model space to find models of high posterior probability.

Many traditional search methods are greedy algorithms, such as stepwise regression. Neural networks are highly nonlinear and often have many local maxima in the model space, and thus greedy algorithms frequently get stuck in local maxima, not finding the optimal model(s). Stochastic algorithms have a clear advantage here. This section presents an algorithm, Bayesian Random Searching (BARS), which is motivated by Markov chain Monte Carlo Model Composition (MC^3) (Raftery, Madigan, and Hoeting 1997).

The basic idea of MC^3 is to perform MCMC on the model space (instead of the parameter space, as usual), thus estimating the posterior probabilities of the models by the fraction of time the chain spends visiting each model. This can be accomplished using the Metropolis algorithm to move between different models with candidates being generated to have either one more or one less explanatory variable, or one more or one less hidden node. The probability of moving from the current model to the candidate model is then the ratio of the posterior probability of the candidate model to the posterior probability of the current model (or one, which ever is smaller). With the BIC approximation, the probability of moving from model i to model j is $\min\{1, \exp[BIC_j - BIC_i]\}$. This BIC approximation also means that one can fit the model with a standard algorithms and even within the Bayesian setting MCMC is not necessary for the parameters during the model search stage. After a best model or set of best models has been found, one can go back and run a full MCMC for the parameters of these models if desired.

While MC^3 may be simulation-consistent, it is not clear how long the simulation needs to run in order to reach equilibrium. Furthermore, the BICs of the models visited are computed, but not used directly in the final estimation of posterior probabilities. Rather than discarding this information and relying solely on the steady state properties of the chain, one could use the same Markov chain simulation, but record the BICs for all of the models visited. Then the estimate of the posterior probability of each model, would be the BIC approximation of Equation 2 if the model was visited, and zero if it was not visited. In practice, this procedure may be more accurate than MC^3 because it does not rely on steady state properties of the chain. The chain is simply a mechanism for effectively searching the model space to find models with high posterior probability. I shall refer to this method as Bayesian Random Searching (BARS). This approach is similar to that of Chipman, George, and McCulloch (1998). BARS can either provide a single "best" model, or posterior probabilities of all models, which can then be used for model averaging.

Neural networks have an additional modelling complication which can affect the search of the model space. In many statistical applications, one can either fit the model analytically, or one can use iterative methods with reasonable confidence of finding the right answer (e.g. logistic regression). However, fitting a neural network, in either a frequentist or Bayesian framework, involves the use of an iterative algorithm which could find a local rather than a global maximum. One should keep in mind that a model visited during a search algorithm may not necessarily be fit correctly. This is a further advantage for stochastic search algorithms such as BARS and MC^3, in that they get multiple chances to fit each model, and so these algorithms are more robust with respect to difficulties in fitting an individual model. They also tend to spend more of their time in areas of the model space with high posterior probability, thus using computational time more efficiently.

3.2 Interpretation of BARS

The derivation of BARS given in this paper is that of a Bayesian method with a noninformative prior over the model space and using the BIC to approximate model probabilities. It could easily be extended to use any reasonable prior over the model space, and thus BARS is seen to be an approximation of a fully Bayesian method.

However, BARS should appeal to the non-Bayesian as well. While inspired by Bayesian methods, one will notice that it is not necessary to ever choose a prior to implement BARS. The BIC is merely a penalized likelihood statistic. Only the maximum likelihood estimate is necessary to compute the BIC. The searching of the model space via BARS can been viewed as a stochastic algorithm based on this penalized likelihood criterion. One need not make any Bayesian assumptions, nor subscribe to the Bayesian philosophy, in order to use BARS.

4 Pima Indians Example

Ripley (1996) provides an analysis of a dataset on diabetes in Pima Indian women. The idea is to predict the presence of diabetes using seven covariates: number of pregnancies, plasma glucose concentration, diastolic blood pressure, triceps skin fold thickness, body mass index (weight/height^2), diabetes pedigree, and age. There are 532 complete records, of which 200 are used as a training set and the other 332 are used as a test set. About 33% of the population has diabetes.

BARS finds that the best model uses only one hidden node and four explanatory variables: number of pregnancies, plasma glucose, body mass, and pedigree. This model has nearly all of the posterior probability, so model averaging is not necessary. The error rate on the test set with this model is 65 of 332, or 19.6%.

Ripley (1996) reports error rates for a number of other analyses. These are summarized in Table 1. For those methods that had model selection techniques readily available (e.g. AIC for logistic regression), the same four variables were found to be important. Notice that the best of the models Ripley investigated was logistic regression. The error rate is essentially the same as that for the neural network found with BARS (the .3% difference is a single test case). This is no coincidence, in that a neural network of the form in this paper with only one hidden node is equivalent to logistic regression. Thus it is reassuring that BARS finds the optimal known answer in this case. We can use the structure of neural networks with all of their flexibility, but BARS will still pick the simplest model when it is most appropriate.

5 Bank Applications Example

Traditionally when customers apply for a loan at a bank, they had to fill out a detailed application. This application was then reviewed by a trained loan officer and the loan was either approved or denied. A particular regional bank wanted to streamline this process to see what information on the application was most important, so that they could greatly simply the form, making life easier for the customer and creating cost savings for the bank.

Standard linear discrimination	20.2%
Robust linear discrimination	22.9%
Logistic regression	19.9%
Multivariate adaptive regression splines (MARS)	22.6%
Projection pursuit regression (PPR)	22.6%
Multi-layer neural network	22.6%
Nearest neighbor with CV	24.7%
Classification tree	24.4%
Neural network with BARS (this paper)	19.6%

Table 1: Comparison of Error Rates on the Pima Indian Diabetes Data

This dataset was originally described in Lee (1996). It involves 23 covariates which are listed in Appendix 1. These variables fall mainly into three groups: stability, demographic, and financial. Stability variables include items such as the length of time the applicant has working in their current job. Stability is thought to be positively correlated with intent and ability to repay a loan, for example, a person who has held their job for longer is less likely to lose their job and thus lose their income and be unable to repay a loan. A person who has lived in their current residence for longer is less likely to suddenly skip town and leave an unpaid loan. Demographic variables include items like age. In the United States, it is illegal to discriminate against older people, but younger people can be discriminated against. Many standard demographic variables (e.g. gender) are not legal for use in a loan decision process and are thus not included. Financial variables include the number of other accounts at this bank and at other financial institutions, as well as the applicant's income and budgeted expenses.

An interesting aspect of this dataset is the large amount of correlation between the covariates. In some cases, there is even causation. For example, someone with a mortgage will be a homeowner. Another example is that a person cannot have lived in their current residence for more years than they are old. Thus any statistical analysis must take this correlation into account, and model selection is an ideal approach.

For this paper, the dataset was split into a training set of 4000 observations and a test set of the remaining 4508 observations. Using the training set, BARS finds that the optimal model uses seven explanatory variables and two hidden nodes. The important variables are income, budgeted expenses, age, length of time at current residence, checking accounts with this bank, accounts at finance companies (typically these companies service customers who have trouble getting loans from standard banks and savings and loans), and category of loan. These are all reasonable variables, and between them they seem to cover most of the important aspects of the covariates without much repetition, thus reducing the multicollinearity problems. This model has an error rate of 31% on the test data, which is not great, but the data are very messy, and there are a number of more subjective factors known to influence loan officers which are not coded in the data, so one can not hope for too precise a fit on this dataset. This model is an improvement over a previous analysis of this data (Lee, 1996) which used model selection for logistic regression, and found the best logistic regression models to use 10-12 covariates and have error rates of around 35%.

6 Conclusions

BARS is a powerful method for model selection that works well for neural networks. It allows the statistician to take advantage of the full flexibility of neural networks while controlling for overfitting. The examples of this paper show that BARS can pick a simple model when necessary, or a richer model when the underlying problem is more complicated. BARS can be interpreted from either a frequentist or Bayesian perspective and is reasonably easy to implement with the BIC approximation.

7 Appendix

Variables in the Bank Loan Application Dataset

1. Birthdate
2. Length of time in current residence
3. Length of time in previous residence
4. Length of time at current employer
5. Length of time at previous employer
6. Line utilization of available credit
7. Number of inquiries on credit file
8. Date of oldest entry in credit file
9. Income
10. Residential status
11. Monthly mortgage payments
12. Number of checking accounts at this bank
13. Number of credit card accounts at this bank
14. Number of personal credit lines at this bank
15. Number of installment loans at this bank
16. Number of accounts at credit unions
17. Number of accounts at other banks
18. Number of accounts at finance companies
19. Number of accounts at other financial institutions
20. Budgeted debt expenses
21. Amount of loan approved
22. Loan type code
23. Presence of a co-applicant

References

Akaike, H. (1974). "A New Look at Statistical Model Identification." *IEEE Transactions on Automatic Control*, AU–19, 716–722.

Andrieu, C., de Freitas, J. F. G., and Doucet, A. (1999). "Robust Full Bayesian Learning for Neural Networks." Tech. Rep. 343, Cambridge Univeristy, Engineering Department.

Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*. Oxford: Clarendon Press.

Bridle, J. S. (1989). "Probabilistic Interpretation of Feedforward Classification Network Outputs, with Relationships to Statistical Pattern Recognition." In *Neuro-computing: Algorithms, Architectures and Applications*, eds. F. Fouglemann-Soulie and J. Héault. New York: Springer-Verlag.

Chipman, H., George, E., and McCulloch, R. (1998). "Bayesian CART Model Search (with discussion)." *Journal of the American Statistical Association*, 93, 935–960.

Cybenko, G. (1989). "Approximation by Superpositions of a Sigmoidal Function." *Mathematics of Control, Signals and Systems*, 2, 303–314.

Funahashi, K. (1989). "On the Approximate Realization of Continuous Mappings by Neural Networks." *Neural Networks*, 2, 3, 183–192.

Hornik, K., Stinchcombe, M., and White, H. (1989). "Multilayer Feedforward Networks are Universal Approximators." *Neural Networks*, 2, 5, 359–366.

Kass, R. E. and Raftery, A. E. (1995). "Bayes Factors." *Journal of the American Statistical Association*, 90, 430, 773–795.

Kass, R. E. and Wasserman, L. (1995). "A Reference Bayesian Test for Nested Hypotheses and Its Relationship to the Schwarz Criterion." *Journal of the American Statistical Association*, 90, 928–934.

Keribin, C. (1997). "Consistent Estimation of the Order of Mixture Models." Tech. rep., Université d'Evry-Val d'Essonne, Laboratoire Analyse et Probabilité.

Leamer, E. E. (1978). *Specification Searches: Ad Hoc Inference with Nonexperimental Data*. New York: Wiley.

Lee, H. K. H. (1996). "Model Selection for Consumer Loan Application Data." Tech. Rep. 650, Carnegie Mellon University, Department of Statistics.

— (1998). "Model Selection and Model Averaging for Neural Networks." Ph.D. thesis, Carnegie Mellon University, Department of Statistics.

— (2000). “Consistency of Posterior Distributions for Neural Networks.” *to appear in Neural Networks*.

MacKay, D. J. C. (1992). “Bayesian Methods for Adaptive Methods.” Ph.D. thesis, California Institute of Technology.

— (1994). “Bayesian Non-Linear Modeling for the Energy Prediction Competition.” *ASHRAE Transactions*, 100, pt. 2, 1053–1062.

McCullagh, P. and Nelder, J. A. (1983). *Generalized Linear Models*. London: Chapman & Hall.

Müller, P. and Rios Insua, D. (1998). “Feedforward Neural Networks for Nonparametric Regression.” In *Practical Nonparametric and Semiparametric Bayesian Statistics*, eds. D. Dey, P. Müller, and D. Sinha. New York: Springer-Verlag.

Murata, N., Yoshizawa, S., and Amari, S. (1994). “Network Information Criterion—Determining the Number of Hidden Units for an Artificial Neural Network Model.” *IEEE Transactions on Neural Networks*, 5, 6, 865–871.

Neal, R. M. (1996). *Bayesian Learning for Neural Networks*. New York: Springer.

Raftery, A. (1996). “Approximate Bayes Factors and Accounting for Model Uncertainty in Generalized Linear Models.” *Biometrika*, 83, 251–266.

Raftery, A. E., Madigan, D., and Hoeting, J. A. (1997). “Bayesian Model Averaging for Linear Regression Models.” *Journal of the American Statistical Association*, 437, 179–191.

Ripley, B. D. (1996). *Pattern Recognition and Neural Networks*. Cambridge: Cambridge University Press.

Schwarz, G. (1978). “Estimating the Dimension of a Model.” *The Annals of Statistics*, 6, 2, 461–464.

Stone, M. (1974). “Cross-validatory Choice and Assessment of Statistical Predictions.” *Journal of the Royal Statistical Society B*, 36, 111–147.